

Data Project No 2 - Optimal Trading Rule for Model Scenarios:

Name: Vignesh Iyer

Collaborators: Bruce Englert, Madison Shaerr, Sarah Marshall, Zach Shepelak

March 26, 2019

1. Optimal Trading Rule for the Following Model Scenarios:

Let M be the number of paths, and n be the number of steps. Fix $N = 100$ steps and $M = 10^6$ paths.

Let ϕ is the ratio/relationship between the co variance to the variance (σ). To show this:

$$\begin{aligned} \text{We know } \tau &= \frac{1}{\kappa} \text{ where } \kappa \text{ is the speed of the mean reversion.} \\ &\implies \phi = e^{-\kappa \Delta t} \text{ where } 0 \leq \phi \leq 1 \\ \text{To show } \kappa &\text{ is always positive, } \tau \geq 0 \implies \kappa \geq 0 \end{aligned}$$

The parameter ϕ is given by the expression

$$\phi = e^{-\kappa \Delta t}$$

where κ is the speed of mean reversion and Δt is the length of a given time period. Thus, ϕ gives how fast the mean will revert.

The parameter ϕ is always less than one. Since $\tau = \frac{1}{\kappa}$ where τ is half-life and κ is the speed of mean reversion and $\tau > 0$ and $\kappa > 0$. Then,

$$\phi = e^{-\kappa \Delta t} < e^0 = 1$$

Therefore, $\phi < 1$.

The parameter m is the expected value of P_n (the long term goal).

Let parameter σ denote the degree of volatility.

It can be assumed that $P_0 = 0$. This is because the purpose of this simulation is to determine the optimal trading rule (i.e. when to exit the market) which gives a profit-taking and stop-loss prices which can be made relative to the initial price by a simple transformation from the initial price, if it is not equal to zero, to zero.

(a) $(\phi, m, \sigma) = (2^{-0.2}, 0, 1) =$

- $(\bar{\pi}, \underline{\pi}) = (0, -10)$
Sharpe ratio : 1.249703
- $(\bar{\pi}, \underline{\pi}) = (2, -10)$
Sharpe ratio : 3.449742
- $(\bar{\pi}, \underline{\pi}) = (4, -10)$
Sharpe ratio : 0.865372
- $(\bar{\pi}, \underline{\pi}) = (0, -5)$
Sharpe ratio : 0.592616
- $(\bar{\pi}, \underline{\pi}) = (2, -5)$
Sharpe ratio : 0.840175
- $(\bar{\pi}, \underline{\pi}) = (4, -5)$
Sharpe ratio : 0.306523
- $(\bar{\pi}, \underline{\pi}) = (0, 0)$
Sharpe ratio : -0.000362
- $(\bar{\pi}, \underline{\pi}) = (2, 0)$
Sharpe ratio : -0.099852
- $(\bar{\pi}, \underline{\pi}) = (4, 0)$
Sharpe ratio : -0.370172

(b) $(\phi, m, \sigma) = (2^{-0.2}, 5, 1) =$

- $(\bar{\pi}, \underline{\pi}) = (0, -10)$
Sharpe ratio : 1.446007
- $(\bar{\pi}, \underline{\pi}) = (2, -10)$
Sharpe ratio : 4.715719
- $(\bar{\pi}, \underline{\pi}) = (4, -10)$
Sharpe ratio : 8.759509
- $(\bar{\pi}, \underline{\pi}) = (0, -5)$
Sharpe ratio : 1.439146
- $(\bar{\pi}, \underline{\pi}) = (2, -5)$
Sharpe ratio : 4.697390
- $(\bar{\pi}, \underline{\pi}) = (4, -5)$
Sharpe ratio : 8.678915
- $(\bar{\pi}, \underline{\pi}) = (0, 0)$
Sharpe ratio : 0.648277
- $(\bar{\pi}, \underline{\pi}) = (2, 0)$
Sharpe ratio : 0.856225
- $(\bar{\pi}, \underline{\pi}) = (4, 0)$
Sharpe ratio : 0.923875

(c) $(\phi, m, \sigma) = (2^{-0.2}, -5, 1) =$

- $(\bar{\pi}, \underline{\pi}) = (0, -10)$
Sharpe ratio : -0.656561
- $(\bar{\pi}, \underline{\pi}) = (2, -10)$
Sharpe ratio : -1.726832
- $(\bar{\pi}, \underline{\pi}) = (4, -10)$
Sharpe ratio : -2.098658
- $(\bar{\pi}, \underline{\pi}) = (0, -5)$
Sharpe ratio : -0.935412
- $(\bar{\pi}, \underline{\pi}) = (2, -5)$
Sharpe ratio : -3.498848
- $(\bar{\pi}, \underline{\pi}) = (4, -5)$
Sharpe ratio : -9.746429
- $(\bar{\pi}, \underline{\pi}) = (0, 0)$
Sharpe ratio : -0.647277
- $(\bar{\pi}, \underline{\pi}) = (2, 0)$
Sharpe ratio : -1.165460
- $(\bar{\pi}, \underline{\pi}) = (4, 0)$
Sharpe ratio : -1.426939

(d) $(\phi, m, \sigma) = (2^{-0.2}, 0, 2) =$

- $(\bar{\pi}, \underline{\pi}) = (0, -10)$
Sharpe ratio : 0.597970
- $(\bar{\pi}, \underline{\pi}) = (2, -10)$
Sharpe ratio : 0.843386
- $(\bar{\pi}, \underline{\pi}) = (4, -10)$
Sharpe ratio : 0.835088
- $(\bar{\pi}, \underline{\pi}) = (0, -5)$
Sharpe ratio : 0.151903
- $(\bar{\pi}, \underline{\pi}) = (2, -5)$
Sharpe ratio : 0.158876
- $(\bar{\pi}, \underline{\pi}) = (4, -5)$
Sharpe ratio : 0.071622
- $(\bar{\pi}, \underline{\pi}) = (0, 0)$
Sharpe ratio : 0.000486
- $(\bar{\pi}, \underline{\pi}) = (2, 0)$
Sharpe ratio : -0.027080
- $(\bar{\pi}, \underline{\pi}) = (4, 0)$
Sharpe ratio : -0.100562

(e) $(\phi, m, \sigma) = (2^{-0.1}, 0, 1) =$

- $(\bar{\pi}, \underline{\pi}) = (0, -10)$
Sharpe ratio : 1.088742
- $(\bar{\pi}, \underline{\pi}) = (2, -10)$
Sharpe ratio : 1.886867
- $(\bar{\pi}, \underline{\pi}) = (4, -10)$
Sharpe ratio : 0.944388
- $(\bar{\pi}, \underline{\pi}) = (0, -5)$
Sharpe ratio : 0.258660
- $(\bar{\pi}, \underline{\pi}) = (2, -5)$
Sharpe ratio : 0.339284
- $(\bar{\pi}, \underline{\pi}) = (4, -5)$
Sharpe ratio : 0.152612
- $(\bar{\pi}, \underline{\pi}) = (0, 0)$
Sharpe ratio : -0.000152
- $(\bar{\pi}, \underline{\pi}) = (2, 0)$
Sharpe ratio : -0.049460
- $(\bar{\pi}, \underline{\pi}) = (4, 0)$
Sharpe ratio : -0.171229

Analysis:

By analyzing the OTR simulation from the test parameter sets (a), (b) and (c), we observe that a higher m resulted in much higher Sharpe ratios, and likewise as predicted a lower m resulted in lower Sharpe ratios. This simply demonstrates that a higher expected value for future price will produce odds in favor of high returns in the tested model.

The tested parameter set (d) differed from set (a) only in that its σ value was greater. As predicted, a higher degree of volatility resulted in a lower Sharpe ratio. Parameter set (e) differed from set (a) in that its ϕ was greater. This too resulted in a lesser Sharpe ratio from that of set (a), which would postulate that a higher rate of mean reversion may yield marginal returns.

Notably, all of the parameter sets with an m of 0 (namely sets (a), (d) and (e)) had a maximum stop-loss value of -10 and optimal profit-taking value of 2. The largest calculated Sharpe ratio was 8.759509, which was for the optimal trading rule of parameter set (b), with a profit-taking value of 4 and a stop-loss value of -10. Note that parameter set (b) had an m of 5, which was the largest parameter given for m .

For parameter set (c), both the maximum stop-loss and profit-taking values were 0. This tested parameter set's trading rule had a Sharpe ratio of -0.647277 , which was the lowest and only negative Sharpe ratio of any of the optimal trading rules. Although not demonstrated here, the use of the Sharpe ratio can be misleading such that negative returns can be brought closer to 0 by either increasing the return or increasing the volatility.

The program was written in Java and source code is provided on the following pages.

```

import java.util.Random;
import java.util.Scanner;

public class Project {

    public static void main (String [] args) {
        Scanner in = new Scanner(System.in);

        //Enter M
        //print("enter the number of paths:");
        //double M = in.nextDouble();
        double M = Math.pow(10, 6);

        //Enter N
        //print("enter the number of steps:");
        //double N = in.nextDouble();
        double N = 100;

        //Enter PU
        print("enter the profit-taking bound: ");
        double PU = in.nextDouble();

        //Enter P1
        print("enter the stop-loss bound: ");
        double PL = in.nextDouble();

        //Enter P0
        print("enter P0: ");
        double P0 = in.nextDouble();

        //Enter mu
        print("enter mu: ");
        double mu = in.nextDouble();

        //Enter tau
        print("enter tau: ");
        double tau = in.nextDouble();

        //Enter sigma
        print("enter sigma: ");
        double sigma = in.nextDouble();

        in.close();

        //Setup phi

```

```

//phi = 2^(-1/tau);
double phi = Math.pow(2, (-1.0/tau));

//Create functions for Price and PnL
double [][] Price = genprice(M, N);
double [][] PnL = genPnL(M);

//Set up ETA
double [][] ETA = genETA(sigma, M, N);

//generate OU paths:

for(int row = 0; row < (int)M; row++) {
for(int col = 1; col < (int)N; col++) {
Price[row][col+1] = mu + phi*(Price[row][col] - mu) + ETA[row][col];
}
}

for(int row = 1; row < (int)M; row++) {
int col = 1;
while(col <= (int) N) {
col = col + 1;
if(Price[row][col] > PU || Price[row][col] < PL) {
break;
}
}
PnL[row][0] = Price[row][col] - Price[row][1];
}

//Generate mean of PnL
double mean = getMean(PnL, (int) M);

//Generate variance of PnL
double variance = getVariance(PnL, (int) M, mean);

Double Sharpe = mean / Math.sqrt(variance);

print("Sharpe ratio for this rule is %8.6f");
print(Sharpe.toString());

}

```

```

/**
 * Calculates the variance of a row vector and returns that value.
 *
 * @param mat
 * @param rows
 * @param mean
 * @return
 */
public static double getVariance(double [][] mat, int rows, double mean) {
    double sum = 0.0;
    double variance = 0.0;

    //Subtract mean and square
    for(int r = 0; r < rows; r++) {
        mat[r][0] -= mean;

        mat[r][0] *= mat[r][0];
    }

    //Sum of the squares
    for(int r = 0; r < rows; r++) {
        sum += mat[r][0];
    }

    variance = sum / (double) rows;

    return variance;
}

/**
 * Calculate Mean of row vector .
 *
 * @param mat
 * @return
 */
public static double getMean(double [][] mat, int rows) {
    double mean = 0.0;
    double sum = 0;

    for(int r = 0; r < rows; r++) {
        sum += mat[r][0];
    }

    mean = sum / (double)rows;

    return mean;
}

```



```

}

/**
 * Generates random array of M by N * sigma
 * @param sigma
 * @param M
 * @param N
 * @return
 */
public static double[][] genETA(double sigma, double M, double N){
    Random rg = new Random(1245);

    double [][] rMat = new double[(int) M][(int)N];
    for(int row = 0; row < (int)M; row ++){
        for(int col = 0; col < (int)N; col++){
            rMat[row][col] = (double)(rg.nextInt((int)M)) * sigma;
        }
    }

    return rMat;
}

/**
 * Returns a zero matrix that is M by 1
 * @param M
 * @return
 */
public static double[][] genPnL(double M){
    return new double[(int)M][1];
}

/**
 * Returns a zero matrix of doubles that is m by n+1
 *
 * @param m
 * @param n
 * @return
 */
public static double[][] genprice(double M, double N){
    return new double[(int)M][(int)(N+1)];
}

```

```
/**
 * Shortcut for printing to the console
 *
 * @param s
 */
public static void print(String s) {
    System.out.println(s);
}
}
```

The following graphs represent the relationship between the upper and lower profit bounds, $(\bar{\pi}, \underline{\pi})$, with the calculated Sharpe ratio for tested parameter sets (a), (b), (c), (d), and (e).

